

Migrating from Previous Versions to Cora SeQuence V9.x

Last Modified on 03/30/2021 9:51 am EDT

V9.x

Overview

Cora SeQuence V9.0 architecture has been restructured into independent modules. These modules are building blocks for building applications, which you can deploy separately. As a result of these structural changes, it is not possible to upgrade Cora SeQuence directly from previous versions to V9.x.

You first need to upgrade your database to V8.8, and only then you can follow a migration path, using SQL scripts to migrate data from your previous database and performing the V9.x application deployment.

NOTE

Cora SeQuence V9.0 application deployment requires fresh servers.

Migration flow

Main steps required to migrate your system.

	Migration step	Details and links to documentation
1	Migration readiness	<p>Make sure that your system is ready for migration. For more details, see this article.</p>
2	Prerequisites	<p>Make sure that:</p> <ul style="list-style-type: none">Your database was upgraded to V8.8.You review all the breaking changes relevant to the version you are upgrading from. For more details, see this article.The application servers are clean servers—servers that did not have Cora SeQuence installed on them previously. <p>NOTE Make sure that the database server has at least the same amount of free disk space as the size of the current Cora SeQuence database.</p>
3	Database migration	Migrate database (see procedure below)
4	V9.x deployment	<ol style="list-style-type: none">Install pre-built applications or Build customized applicationsDeploy applications
5	Post-migration steps	<ul style="list-style-type: none">Delete unused tablesFind activities that use the Shared Resources folderRemove redundancy property from jobs <p>For more details, see this article.</p>

Migrate database

You migrate the database using SQL scripts.

1. [Download](#) the migration scripts.
2. Back up the database.
3. To make sure that applications from the previous version do not connect to the V9x database, make sure that:
 - a. You shut down all the application servers, and leave only the database server on.
 - b. Do *one* of the following:
 - Rename the database with a different name.
 - Restore the database *as a new database* and use the *restored* database with a new name.
4. To avoid a full transaction log, change the database recovery model to**Simple**.
5. Run the following scripts:
 - a. **01 - Pre-Migration script JES.sql**
 - b. **02 - Database.sql**
 - c. **03 - Objects.sql**
 - d. **04 - Data.sql**
 - e. **05 - Fill Global Solution Tables.sql**
 - f. **06 - Consolidate UWF Tables.sql**

NOTE

Depending on the amount of data in the USL and UWF tables, the *Fill Global Solution Tables* and *Consolidate UWF Tables* scripts can take several hours to run.

- g. **07 - JES Migration script.sql**

NOTE

In case there are jobs that should not be redundant, change their redundancy configuration after the migration is completed.

For more details, see the **Next steps** section.

- h. **08 - End Migration script JES.sql**

6. To verify that the database migration was successful, run the following command:

SELECT fldVersion FROM tblCompanyParam

Expected result: **9.0.0.x**

7. If relevant, change the database recovery model back to the required option.

Next steps

After the migration process ends, review the post-migration procedures, to decide if they are relevant for your implementation.

For more details, see [this article](#).

Rollback to previous version

1. Restore the backed up database to previous state.

